(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) Publication number: **0 290 942 A2**

(12) EUROPEAN PATENT APPLICATION

(21) Application number: **88107159.1**

(51) Int. Cl.4: **G06F 9/46**

(22) Date of filing: **04.05.88**

(30) Priority: **11.05.87 JP 114243/87**

(43) Date of publication of application:
17.11.88 Bulletin 88/46

(84) Designated Contracting States:
DE FR GB

(71) Applicant: **HITACHI, LTD.**
6, Kanda Surugadai 4-chome
Chiyoda-ku Tokyo 101(JP)

Applicant: **Hitachi Computer Engineering Co., Ltd.**
1, Horiyamashita
Hadano-shi Kanagawa-ken(JP)

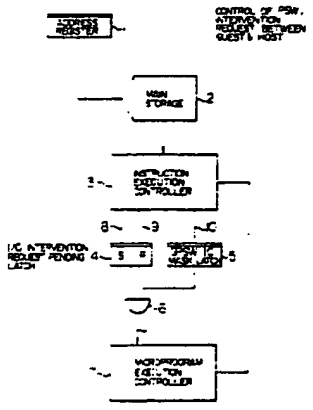
(72) Inventor: **Watanabe, Masaya**
Ishii Haitsu 202, 9-31
Hagigaoka Hadano-shi(JP)
Inventor: **Wakui, Fujio**
Kopo Katakura 103, 35-3
Horikawa Hadano-shi(JP)
Inventor: **Abe, Shuichi**
Soaru Ohatano 901, 1-7
Imagawacho Hadano-shi(JP)

(74) Representative: **Altenburg, Udo, Dipl.-Phys. et al**
Patent- und Rechtsanwälte
Bardehle-Pagenberg-Dost-Altenburg
Frohwitter-Geissler & Partner Galileiplatz 1
Postfach 86 06 20
D-8000 München 86(DE)

(54) Guest machine execution control system for virtual machine system.

(57) Mask data (5) in a PSW are latched and sent from a host to a guest are subjected to a logical AND operation with latched data (4) taken from an intervention request field of a state descriptor (13) of the guest machine. The resultant logical product (6) is used for setting the PSW of the guest machine and activating an interception to the host. The overhead of an execution controller (3) managing the state transition between guest-host machines is reduced.

FIG. 1



EP 0 290 942 A2

GUEST MACHINE EXECUTION CONTROL SYSTEM FOR VIRTUAL MACHINE SYSTEM

BACKGROUND OF THE INVENTION

The present invention relates to a virtual machine system and more particularly to a guest machine execution control system suitable for processing guest instructions at high speed.

As shown in U.S. Patent No. 4,456,954, in the virtual machine system, operation of a virtual machine (hereinafter called VM) starts only after a start interpretative execution instruction (hereinafter called SIE instruction) for indicating execution start of the virtual machine has been executed.

Fig. 2 of the accompanying drawings shows the format of an SIE instruction which designates as operands a state descriptor (hereinafter called SD) on a main storage (hereinafter called MS), by using a base register number field B2 and a displacement field D2. An instruction unit of a central processing unit (hereinafter called CPU) decodes the SIE instruction to read out the respective SD fields from the MS and place the fields in hardware resources of the CPU, thereby permitting the CPU to operate as a guest VM. As the hardware resources in which the SD fields can be placed, there may be mentioned a guest mode latch, a program status word (PSW) register, a control register (CR), a general register (GR), a prefix register, an MS extent register and the like. Thus, the SD describeds the architecture of the guest VM as viewed from the side of a guest instruction. On the other hand, the state of the host is reserved in the CPU.

The guest VM started in response to the SIE instruction continues to emulate a guest program until interruption or interception for control transfer to the VMCP takes place. With the term "interruption", it is intended to mean interruptions such as input/output interruption and external interruption which require the processing by the host. The interception may occur, for example, when an instruction requiring the processing by the host has to be executed. Whether the interception is to be issued or not for some of privileged instructions can be designated by an interception mask field contained in the SD. Upon occurrence of the interruption or interception, operation of the guest VM comes to an end, whereupon the CPU is set to the host mode to be restored to the state ready for executing the SIE instruction.

In a virtual machine system, all or part of the I/O processings by the guest are simulated with the VMCP. Namely, when an I/O activation instruction is designated in a guest program, the guest VM stops its operation and an interception is noticed to the VMCP which then simulates the I/O activation instruction of the guest program to start the I/O processings for a real I/O device. A corresponding I/O interruption instruction is acknowledged by the VMCP according to an I/O mask of the host. The VMCP suspends or holds the guest I/O interruption until the guest I/O mask turns on, whereupon the guest I/O interruption is enabled. Similar processings are performed for some of external interruptions from the guest. In the following description, a term "intervention request" is used for indicating that acknowledgement of a guest interruption by the VMCP has been noticed to the guest and the guest interruption is suspended by the VMCP until the mask of the guest turns on.

In the virtual machine system, an intervention request field is provided in the SD as shown in Fig. 2 for the VMCP to suspend the guest interruption. The intervention request field is constructed of bits each being related to a main cause of an interruption to be suspended. Particularly, it is constructed of an I/O intervention request bit related to an I/O interruption, an external intervention request bit related to an external interruption and the like.

A flow of processing the I/O interruption to be suspended by the VMCP is shown in Fig. 3. At step 28, the VMCP acknowledges an I/O interruption request from a particular guest VM (indicated by guest 1). At step 29, the VMCP sets the I/O intervention request bit, in the intervention request field of the SD, at "1" to suspend the acknowledged guest I/O interruption request. Thereafter, the VMCP continues performing the processings and issues the SIE instruction to activate the guest 1. If the I/O intervention request bit of the SD is "1", the guest 1 stops its operation when the I/O mask (indicated by GPSW(IO)) of the guest PSW becomes "1" and notifies the VMCP of an interception.

In more particular, at step 31 the guest 1 starts executing guest instructions since the GPSW (IO) is 0 in the example shown in Fig. 3. Step 32 indicates that the GPSW (IO) changes from "0" to "1" after execution of a Load PSW (LPSW) instruction by which data read from the main storage at the location designated by the operands of the instruction are loaded in the current PSW register, a Store-Then-OR-System-Mask (STOSM) instruction by which the current system mask is stored in the main storage at the designated location and then data of the instruction are OR-ed with the system mask, a Set-System-Mask (SSM) instruction by which data read from the main storage at the location designated by the operands are set in

the system mask, or other instructions associated with the system mask. The condition of interception is satisfied when such a change occurs, whereupon the guest 1 terminates executing the guest instruction and notifies the VMCP of the interception (step 33). The VMCP simulates the I/O interruption of the guest 1 (step 34) to again activate the guest 1 (step 35). The guest 1 resumes the processings after the interruption by the I/O interruption handler.

The VMCP suspends the guest external interruption in the similar manner as above, so that the following description is directed only to the guest I/O interruption.

As seen from step 32 of Fig. 3, if an instruction changing the system mask of the guest VM is executed and the GPSW (IO) changes from "0" to "1", it becomes necessary to judge if operation of the guest VM is to be terminated upon reception of an interception or otherwise to be continued, based on the presence or absence of the I/O intervention request.

Fig. 4 shows the sequence of processing an LPSW instruction. The process sequence will be described by using an LPSW instruction by way of example, the same sequence being applied to other instructions changing the system mask. It is first checked at step 14 if there is an access exception or a designation exception. If a cause of exception is present, an interruption process is performed at step 23. If not present, a flag is initialized (step 15). If the GPSW (IO) to be set newly is "1" (step 16), the intervention request field is read from the SD of the guest VM now in concern (step 17). If the I/O request bit of the read-out intervention request field is "1" (step 18), the flag is set at "1" (step 19). The flag "1" indicates that the condition of issuance of an interception in response to the I/O intervention request has been met. Next, the PSW register is set (step 20). If both the flag and the GPSW (IO) are "1", then an interception process is performed (steps 21, 22). If not, processing the LPSW instruction is terminated.

The occurrence frequency of I/O intervention requests is generally so small that the performance to be achieved by the system takes into consideration the case where an LPSW instruction is executed normally over a process main path including steps 14, 15, 16, 20, 21 and 37 or a process main path including steps 14, 15, 16, 17, 18, 20, 21 and 37. Steps for reading an intervention request field and checking an I/O intervention request bit are provided in addition to those by the host LPSW instruction. Such additional steps involve the overhead of processing a guest instruction.

According to the technique described as above, each time an instruction which changes the guest system mask is executed, it becomes necessary to check the presence or absence of an I/O intervention request and check if the condition of an interception by the I/O intervention request has been met, by referring to the SD on the main storage (or a temporary copy of the SD on the local storage of the CPU). Thus, it poses the problem of substantial decline of the performance of processing an instruction which changes the guest system mask.

SUMMARY OF THE INVENTION

It is an object of the present invention to provide a guest machine execution control system in a virtual machine system capable of processing a guest instruction which changes the guest system mask, at a speed substantially equal to that of processing a host instruction.

It is another object of the invention to provide a circuit for checking an intervention request set in the VM state descriptor by the VMCP during execution of a guest machine in a virtual machine system.

The above objects are achieved by a circuit comprising a latch group (intervention request pending latch) provided in the CPU for storing the value of intervention request field of the SD of a guest VM under execution, wherein the intervention request pending latch is set when an SIE instruction is executed and renewed at a proper period, and whereby if the intervention request latch is "1" during execution of the guest VM and the bit corresponding to the system mask of the guest VM is on, an interruption is issued to the microprogram of the CPU. After the interruption, the microprogram execution controller causes the execution of the guest VM to stop upon issuance of an interception.

The condition of the intervention request at the VMCP executing a guest VM and the guest system mask is checked in a hardware manner. Therefore, the condition needs not be checked by the microprogram when an instruction which changes the guest system mask is to be executed, to thereby improve the performance of processing an instruction which changes the guest system mask.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a block diagram showing an embodiment of a control system according to the present invention;

Fig. 2 shows the format of a start interpretative execution (SIE) instruction for starting the execution of a virtual machine;

Fig. 3 shows a flow chart illustrating the case where the VMCP suspends a guest interruption;

Fig. 4 is a flow chart showing the processing of an LPSW instruction; and

Fig. 5 is a flow chart showing the processing of an LPSW instruction according to the present invention.

DESCRIPTION OF THE PREFERRED EMBODIMENT

The embodiment of this invention will now be described with reference to the accompanying drawings.

Fig. 1 shows the arrangement of the control system of this invention. In the Figure, reference numeral 1 denotes an address register, 2 a main storage, and 3 an instruction execution controller. Reference numeral 4 denotes an I/O intervention request pending latch, 5 a guest PSWIO mask latch, and 6 an AND gate for obtaining a logical product of the I/O intervention request latch 4 and the guest PSWIO mask latch 5. The elements 4 to 6 are used for a guest I/O interruption. Used for a guest external interruption are similar elements which are omitted in this embodiment. The instruction execution controller 3 causes the I/O intervention request pending latch 4 to set via a control line 8 and reset via a control line 9 to thereby renew the data stored in the latch 4. An output from the AND gate 6 serves as an interruption trigger 11 which notifies a microprogram execution controller 7 of an interruption of the microprogram.

The operation of the control system shown in Fig. 1 will be described with reference to Fig. 3. An SIE instruction is issued at step 30 to activate the guest 1. In processing the SIE instruction, an address designated by the B2 and D2 fields of the SIE instruction is set at the address register 1 to read the SD from the main storage 3. Of the readout SD, the I/O intervention request bit and the guest PSWIO mask are set at the I/O intervention request pending latch 4 and the guest PSWIO mask latch 5, respectively, via the instruction execution controller 3. The I/O intervention pending latch 4 is renewed at a proper period during executing the guest VM, by reading the SD on the main storage 3. This renewal is carried out in order to reflect an I/O intervention request acknowledged and suspended by another CPU in a multi-processor system while the guest 1 is executing its own I/O intervention request. On the other hand, the guest PSWIO mask latch 5 is renewed upon execution of an instruction which changes the guest mask, the instruction including such as an LPSW instruction, STOSM instruction, STNSM instruction, SSM instruction and the like.

If the I/O intervention request pending latch 4 is "1" and the guest PSWIO mask latch 5 is on, then the AND gate 6 outputs an interruption trigger 11 of "1" to notify the microprogram execution controller 7 of an interruption. In more particular, if the VMCP is suspending the guest I/O interruption, the I/O intervention request pending latch 4 becomes "1" so that the guest VM runs while the guest PSWIO mask latch 5 is on. However, immediately after the guest PSWIO mask becomes on (step 32) upon execution of an instruction which changes the guest system mask, the interruption trigger 11 becomes on to terminate the execution of the guest VM at the interception process (step 33) by means of the interruption to the microprogram. If the execution of the guest VM terminates due to other causes, the I/O intervention request pending latch 4 is reset after the termination.

Since the condition of an interception to be caused by the guest interruption suspended by the VMCP is checked during the execution of the guest VM, the condition needs not be checked by the microprogram when a control instruction of the guest VM is to be executed. This operation will then be described with reference to Fig. 5 which shows the processing of an instruction which changes a system mask of the guest PSW, by using an LPSW by way of example. An access exception and a designation exception are checked at step 24. If there is no cause of exception, the PSW is set at step 25 to thereafter terminate executing the instruction. This eliminates the overhead of processing the guest instruction as discussed previously. On the other hand, if there is any cause of exception, an interruption process is performed (step 26). If the interception condition is met after the I/O intervention request, then the microprogram issues an interruption as discussed previously to perform an interception process (step 27). 6

Claims

1. A virtual machine system including a main storage (2) for storing a state descriptor for a guest virtual machine, an instruction execution unit (3) for starting the execution of the guest virtual machine such that in executing an instruction for starting the execution of the virtual machine, the state descriptor for the guest virtual machine on the main storage at the location designated by the instruction is set in a hardware resource, wherein when the state of the guest virtual machine under execution satisfies a predetermined condition, information whether the execution of the guest virtual machine is to be

terminated or not is stored in the state descriptor for the guest virtual machine, said virtual machine system comprising:

(a) first means (4) for storing said information, stored in said state descriptor of the guest virtual descriptor, whether the execution of the guest virtual machine is to be terminated or not;

(b) second means (8, 9) for renewing said first means while at least said instruction for starting the execution of the guest virtual machine is executed.

(c) third means (5) for storing the state of the guest virtual machine with respect to said predetermined condition; and

(d) fourth means (6, 7, 11) for terminating the execution of the guest virtual machine in response to a predetermined relation between outputs from said first and third means.

2. A virtual machine system according to Claim 1, wherein said first means comprises a flag latch circuit for storing a flag which is sent to the guest virtual machine to request an intervention to the guest virtual machine, and said instruction execution unit renews the flag in said latch circuit via said second means.

3. A virtual machine system according to Claim 1, wherein said third means (5) in a mask latch circuit for holding mask data which are used for selective permission of operation interruptions of the guest virtual machine.

4. A virtual machine system according to Claim 1, wherein said fourth means comprises a logical gate circuit (6) for performing an AND operation of the outputs from said flag latch circuit and said mask latch circuit.

5. In a virtual machine system for running a plurality of guest virtual machine under control of a host machine, a control circuit for controlling the state transition between host-virtual machines in accordance with a guest virtual machine state descriptor and mask data for selectively masking a plurality of interruptions for intercepting the execution of the guest virtual machine, respectively set in a hardware resource in response to a start instruction for starting the execution of the guest virtual machine, said control circuit comprising:

(a) a logical product circuit (6) for obtaining a logical product of at least two inputs;

(b) a mask latch circuit (5) connected to one of said at least two inputs for storing said mask data;

(c) an intervention request flag latch circuit (4) connected to another of said at least two inputs for holding a flag representative of an execution intervention to the guest virtual machine taken from an intervention request field contained in said guest virtual machine descriptor: and

(d) an instruction execution unit (3, 7) for performing an intercepting operation from the guest virtual machine to the host machine in accordance with an output from said logical product circuit.

6. A control circuit according to Claim 5, wherein said instruction execution unit comprises means (25) for setting a PSW in response to an instruction which changes a system mask of the guest virtual machine, and interception means (27) for performing an intercepting operation from the guest virtual machine to the host machine by means of an interruption.

7. A control circuit according to Claim 6 comprising means (3) for renewing the mask data in said mask latch circuit in response to any one of the instructions supplied from the host machine, said instructions including a load PSW instruction, a Store-Then-OR-System-Mask instruction, a Store-Then-AND-System-Mask instruction, and a Set-System-Mask instruction.

8. A control circuit according to Claim 7 wherein said start instruction is a start interpretative execution instruction supplied from the host machine, said instruction having a base register number (B2) and a displacement (D2).

FIG. 1

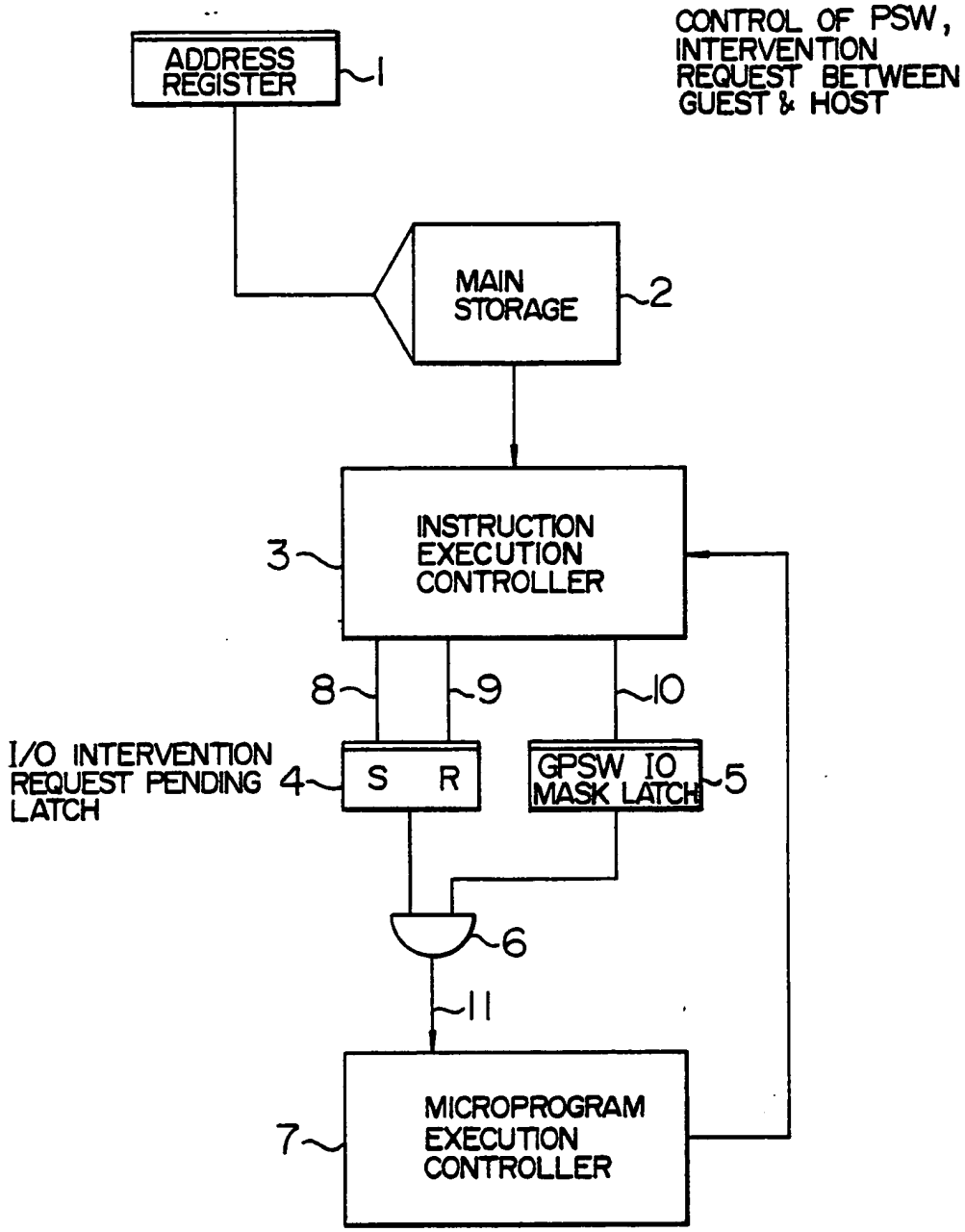

ADDRESS REGISTER
1
CONTROL OF PSW, INTERVENTION REQUEST BETWEEN GUEST & HOST
MAIN STORAGE
2
INSTRUCTION EXECUTION CONTROLLER
3
8
9
10
I/O INTERVENTION REQUEST PENDING LATCH
4
S R
GPSW IO MASK LATCH
5
6
11
MICROPROGRAM EXECUTION CONTROLLER
7

FIG. 2

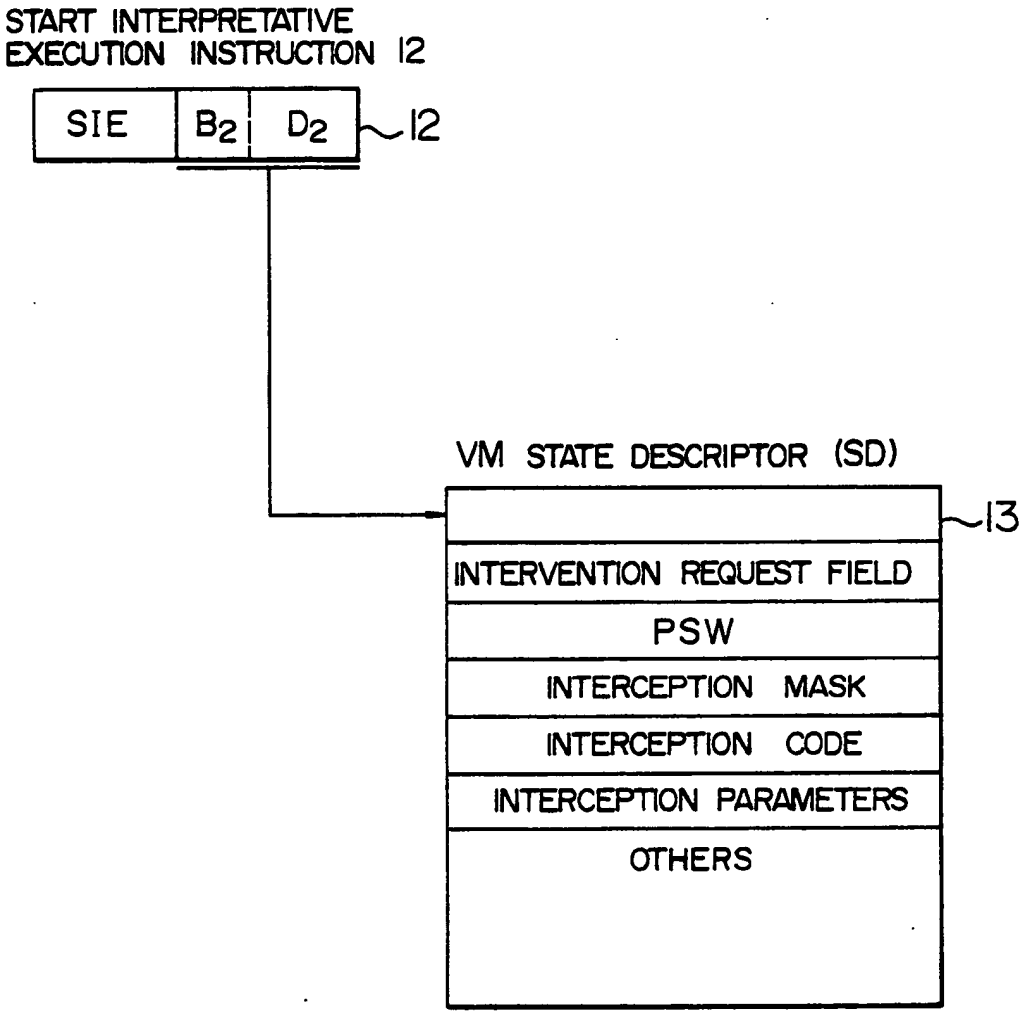

START INTERPRETATIVE
EXECUTION INSTRUCTION 12
SIE
B2
D2
12
VM STATE DESCRIPTOR (SD)
13
INTERVENTION REQUEST FIELD
PSW
INTERCEPTION MASK
INTERCEPTION CODE
INTERCEPTION PARAMETERS
OTHERS

FIG. 3

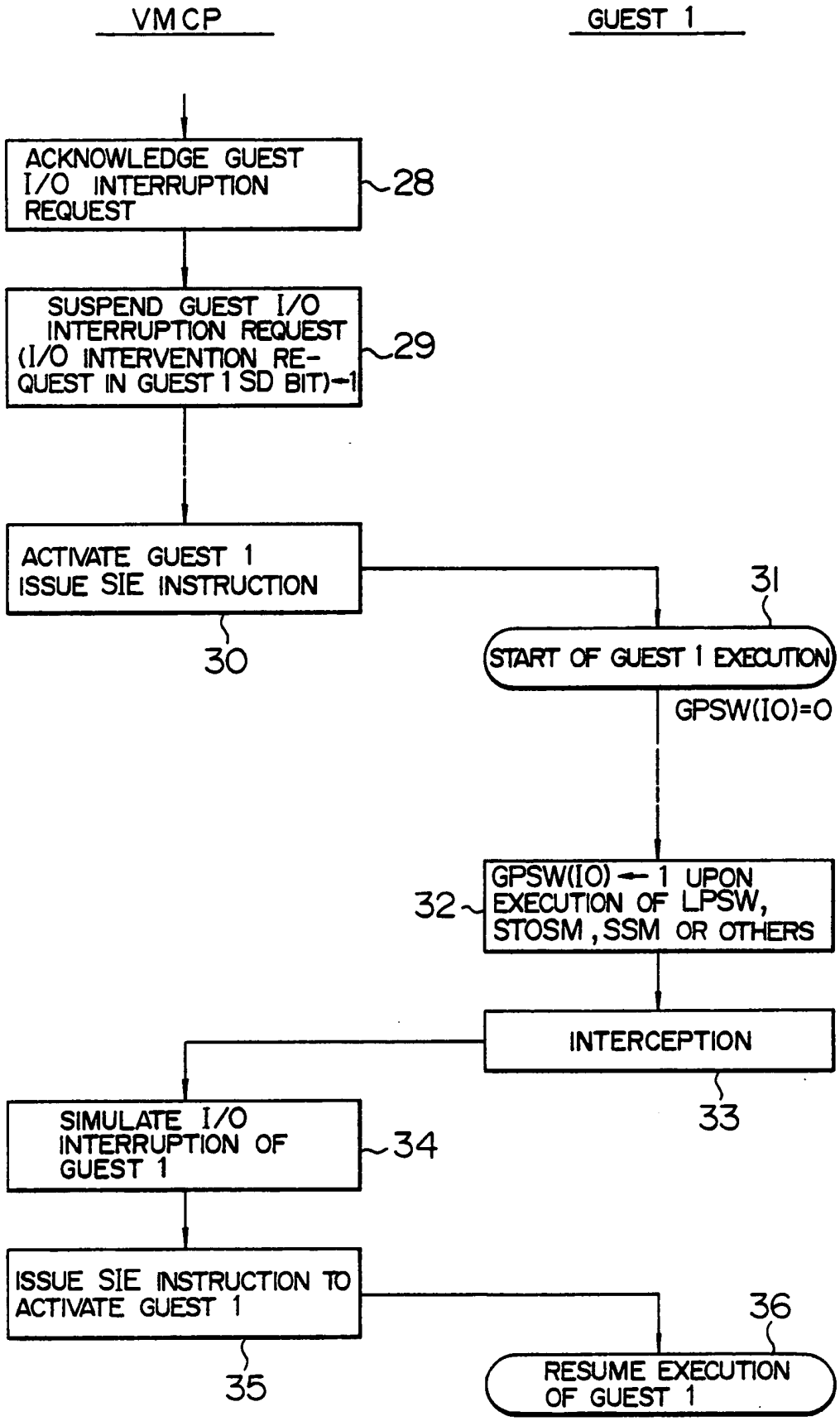

VMCP
GUEST 1
ACKNOWLEDGE GUEST I/O INTERRUPTION REQUEST
28
SUSPEND GUEST I/O INTERRUPTION REQUEST (I/O INTERVENTION REQUEST IN GUEST 1 SD BIT)←1
29
ACTIVATE GUEST 1 ISSUE SIE INSTRUCTION
30
31
START OF GUEST 1 EXECUTION
GPSW(IO)=0
GPSW(IO)←1 UPON EXECUTION OF LPSW, STOSM, SSM OR OTHERS
32
INTERCEPTION
33
SIMULATE I/O INTERRUPTION OF GUEST 1
34
ISSUE SIE INSTRUCTION TO ACTIVATE GUEST 1
35
36
RESUME EXECUTION OF GUEST 1

FIG. 4

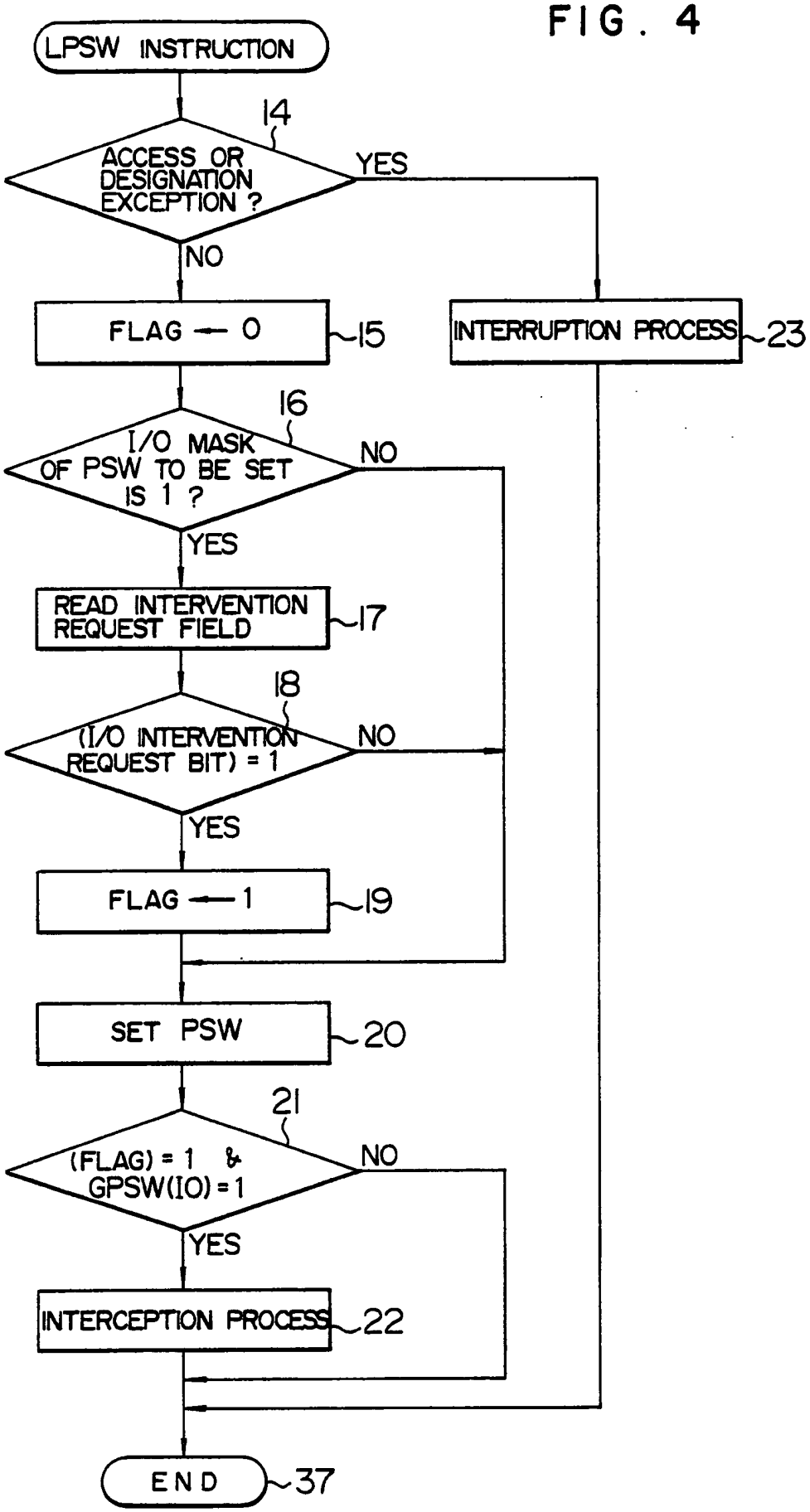
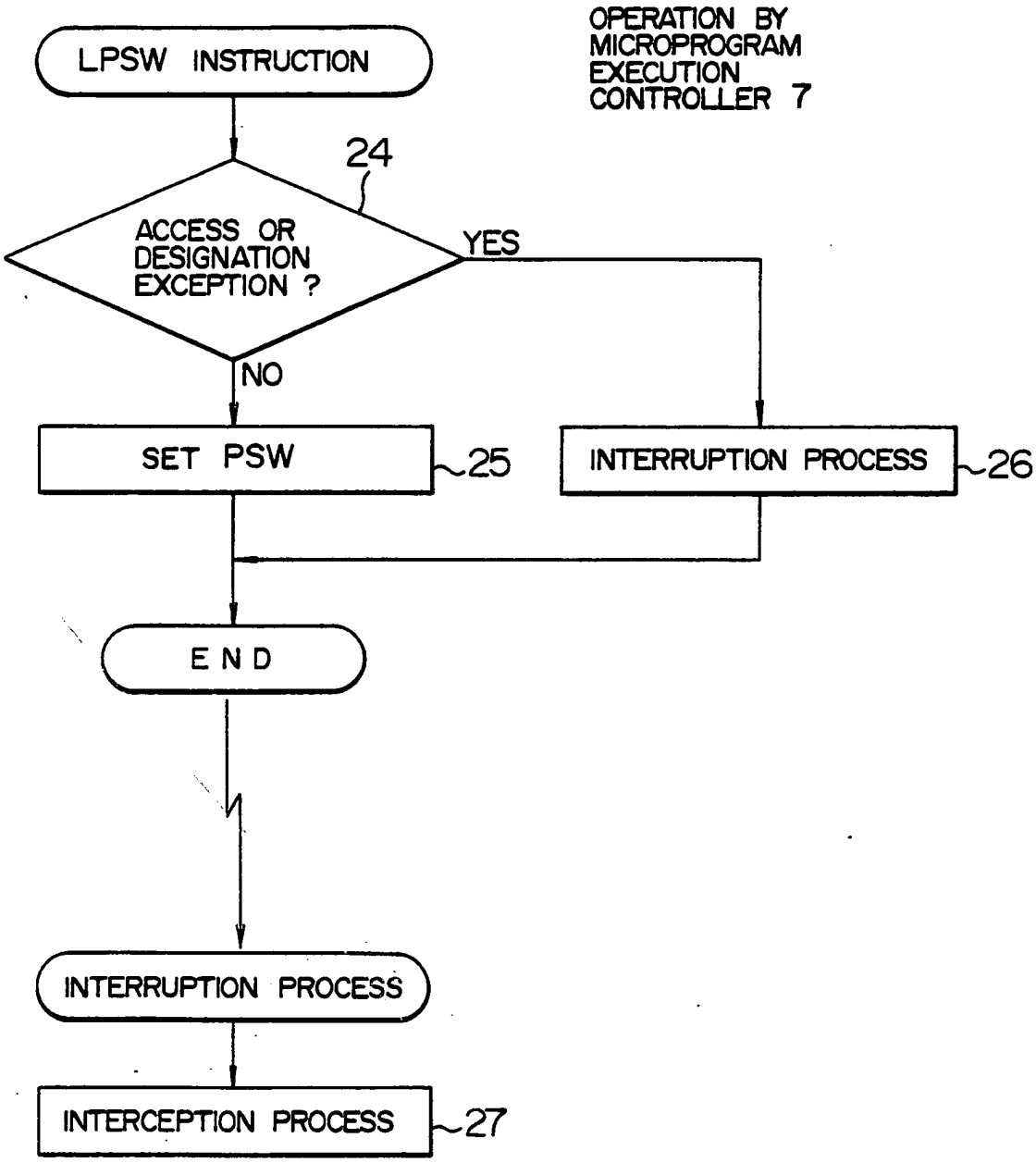

LPSW INSTRUCTION
14
ACCESS OR DESIGNATION EXCEPTION ?
YES
NO
FLAG ← 0
15
INTERRUPTION PROCESS
23
16
I/O MASK OF PSW TO BE SET IS 1 ?
NO
YES
READ INTERVENTION REQUEST FIELD
17
18
(I/O INTERVENTION REQUEST BIT) = 1
NO
YES
FLAG ← 1
19
SET PSW
20
21
(FLAG) = 1 & GPSW(IO) = 1
NO
YES
INTERCEPTION PROCESS
22
END
37

FIG. 5


OPERATION BY MICROPROGRAM EXECUTION CONTROLLER 7
LPSW INSTRUCTION
24
ACCESS OR DESIGNATION EXCEPTION ?
YES
NO
SET PSW
25
INTERRUPTION PROCESS
26
END
INTERRUPTION PROCESS
INTERCEPTION PROCESS
27